Linear Convergent Decentralized Optimization with Compression

Xiaorui Liu
http://cse.msu.edu/~xiaorui/

Joint work with Yao Li, Rongrong Wang, Jiliang Tang, and Ming Yan

Data Science and Engineering Lab
Department of Computer Science and Engineering
Michigan State University

ICLR 2021, May 6th
Introduction

- Problem

\[ x^* := \arg \min_{x \in \mathbb{R}^d} \left[ f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right] \]

- \( f_i(\cdot) \) is the local objective in agent \( i \).

Centralization

\[ f_1(x), f_2(x), \ldots, f_{n-1}(x), f_n(x) \]

Decentralization

\[ f_1(x), f_2(x), f_3(x), f_4(x), f_5(x) \]
Matrix notations

\[ X^k = \begin{bmatrix}
\vdots \\
(\mathbf{x}_1^k)^\top & \vdots \\
(\mathbf{x}_n^k)^\top & \vdots \\
\vdots \\
(\nabla f_1(\mathbf{x}_1^k))^\top & \vdots \\
(\nabla f_1(\mathbf{x}_n^k))^\top & \vdots \\
\end{bmatrix} \in \mathbb{R}^{n \times d}, \]

\[ \nabla \mathbf{F}(X^k) = \begin{bmatrix}
\vdots \\
(\nabla f_1(\mathbf{x}_1^k))^\top & \vdots \\
(\nabla f_1(\mathbf{x}_n^k))^\top & \vdots \\
\vdots \\
\vdots \\
\vdots \\
\end{bmatrix} \in \mathbb{R}^{n \times d}, \]

Symmetric \( \mathbf{W} \in \mathbb{R}^{n \times n} \) encodes the communication network.

\[ \mathbf{W} X = X \text{ iff } \mathbf{x}_1 = \mathbf{x}_2 = \cdots = \mathbf{x}_n, \]

\[ -1 < \lambda_n(\mathbf{W}) \leq \lambda_{n-1}(\mathbf{W}) \leq \cdots \lambda_2(\mathbf{W}) < \lambda_1(\mathbf{W}) = 1. \]
Communication Compression for decentralized optimization
- DCD-SGD, ECE-SGD [TGZ+18]
- QDGD, QuanTimed-DSGD [RMHP19, RTM+19]
- DeepSqueeze [TLQ+19]
- CHOCO-SGD [KSJ19]
- ...

Reduce to DGD-type algorithms, which suffer from convergence bias

\[ \mathbf{X}^* \neq \mathbf{W}\mathbf{X}^* - \eta \nabla \mathbf{F}(\mathbf{X}^*). \]

Their convergences degrade on heterogeneous data.

LEAD is the first primal-dual decentralized optimization algorithm with compression and attains linear convergence.
Algorithm: LEAD

- NIDS [LSY19] / D² [TLY+18] (stochastic version of NIDS)

\[
X^{k+1} = \frac{I + W}{2} \left( 2X^k - X^{k-1} - \eta \nabla F(X^k; \xi^k) + \eta \nabla F(X^{k-1}; \xi^{k-1}) \right),
\]

- A two step reformulation [LY19]:

\[
D^{k+1} = D^k + \frac{I - W}{2\eta} \left( X^k - \eta \nabla F(X^k) - \eta D^k \right),
\]

\[
X^{k+1} = X^k - \eta \nabla F(X^k) - \eta D^{k+1},
\]

- Concise and conceptual form of LEAD:

\[
Y^k = X^k - \eta \nabla F(X^k; \xi^k) - \eta D^k
\]

\[
\hat{Y}^k = \text{CompressionProcedure}(Y^k)
\]

\[
D^{k+1} = D^k + \frac{\gamma}{2\eta} (I - W) \hat{Y}^k
\]

\[
X^{k+1} = X^k - \eta \nabla F(X^k; \xi^k) - \eta D^{k+1}
\]
Algorithm: LEAD

LEAD

\[ Y^k = X^k - \eta \nabla F(X^k; \xi^k) - \eta D^k \]
\[ \hat{Y}^k = \text{CompressionProcedure}(Y^k) \]
\[ D^{k+1} = D^k + \frac{\gamma}{2\eta} (I - W) \hat{Y}^k = \frac{\gamma}{2\eta} (\hat{Y}^k - \hat{Y}^k_w) \]
\[ X^{k+1} = X^k - \eta \nabla F(X^k; \xi^k) - \eta D^{k+1} \]

Compression Procedure

\[ Q^k = \text{Compress}(Y^k - H^k) \quad \triangleright \text{Compression} \]
\[ \hat{Y}^k = H^k + Q^k \]
\[ \hat{Y}^k_w = H^k_w + WQ^k \quad \triangleright \text{Communication} \]
\[ H^{k+1} = (1 - \alpha)H^k + \alpha \hat{Y}^k \]
\[ H^{k+1}_w = (1 - \alpha)H^k_w + \alpha \hat{Y}^k_w \]
Algorithm: LEAD

How LEAD works?

- **Gradient Correction**
  \[
  \mathbf{X}^{k+1} = \mathbf{X}^k - \eta (\nabla \mathbf{F}(\mathbf{X}^k; \xi^k) + \mathbf{D}^{k+1})
  \]
  \[
  \mathbf{F}(\mathbf{X}^k; \xi^k) + \mathbf{D}^{k+1} \to 0
  \]

- **Difference Compression**
  \[
  \mathbf{Q}^k = \text{Compress}(\mathbf{Y}^k - \mathbf{H}^k)
  \]
  \[
  \mathbf{Y}^k \to \mathbf{X}^*, \mathbf{H}^k \to \mathbf{X}^* \Rightarrow \mathbf{Y}^k - \mathbf{H}^k \to 0 \Rightarrow \|\mathbf{Q}^k - (\mathbf{Y}^k - \mathbf{H}^k)\| \to 0
  \]

- **Implicit Error Compensation**
  \[
  \mathbf{E}^k = \hat{\mathbf{Y}}^k - \mathbf{Y}^k
  \]
  \[
  \mathbf{D}^{k+1} = \mathbf{D}^k + \frac{\gamma}{2\eta} (\hat{\mathbf{Y}}^k - \hat{\mathbf{Y}}_w^k) = \mathbf{D}^k + \frac{\gamma}{2\eta} (\mathbf{I} - \mathbf{W})\mathbf{Y}^k + \frac{\gamma}{2\eta} (\mathbf{E}^k - \mathbf{W}\mathbf{E}^k)
  \]
Assumption

- Compression: $\mathbb{E} Q(x) = x$, $\mathbb{E} \| x - Q(x) \|^2 \leq C \| x \|^2$ for some $C \geq 0$.
- $f_i(\cdot)$ is $\mu$-strongly convex and $L$-smooth:
  \[
  f_i(x) \geq f_i(y) + \langle \nabla f_i(y), x - y \rangle + \frac{\mu}{2} \| x - y \|^2,
  \]
  \[
  f_i(x) \leq f_i(y) + \langle \nabla f_i(y), x - y \rangle + \frac{L}{2} \| x - y \|^2.
  \]
- Gradient: $\mathbb{E}_{\xi} \nabla f_i(x; \xi) = \nabla f_i(x)$, $\mathbb{E}_{\xi} \| \nabla f_i(x; \xi) - \nabla f_i(x) \|^2 \leq \sigma^2$. 
\[ \kappa_f = \frac{L}{\mu}, \quad \kappa_g = \frac{\lambda_{\max}(I - W)}{\lambda_{\min}(I - W)} \]

**Theorem (Complexity bounds when \( \sigma = 0 \))**

- **LEAD converges to the \( \epsilon \)-accurate solution with the iteration complexity**
  \[ O\left( \left( (1 + C)(\kappa_f + \kappa_g) + C\kappa_f\kappa_g \right) \log \frac{1}{\epsilon} \right). \]

- **When \( C = 0 \) (i.e., no compression) or \( C \leq \frac{\kappa_f + \kappa_g}{\kappa_f\kappa_g + \kappa_f + \kappa_g} \), the iteration complexity is**
  \[ O\left( (\kappa_f + \kappa_g) \log \frac{1}{\epsilon} \right). \]

  *This recovers the convergence rate of NIDS [LSY19].*
Theorem (Complexity bounds when $\sigma = 0$)

- With $C = 0$ (or $C \leq \frac{\kappa_f + \kappa_g}{\kappa_f \kappa_g + \kappa_f + \kappa_g}$) and fully connected communication graph (i.e., $W = \frac{11^T}{n}$), the iteration complexity is

$$\mathcal{O}(\kappa_f \log \frac{1}{\epsilon}).$$

This recovers the convergence rate of gradient descent [Nes13].

Theorem (Error bound when $\sigma > 0$)

$$\frac{1}{n} \sum_{i=1}^{n} \mathbb{E} \left\| x_i^k - x^* \right\|^2 \lesssim \mathcal{O}\left( \frac{1}{k} \right)$$
Linear regression ($\sigma = 0$)
Experiment

Consensus error

Compression error

Linear regression ($\sigma = 0$)
Logistic regression ($\sigma > 0$).
Loss $f(X^k)$

Stochastic optimization on deep learning (* divergence).
Conclusion

- LEAD is the first primal-dual decentralized optimization algorithm with compression and attains linear convergence for strongly convex and smooth objectives.
- LEAD supports unbiased compression of arbitrary precision.
- LEAD works well for nonconvex problems such as training deep neural networks.
- LEAD is robust to parameter settings, and needs minor effort for parameter tuning.

Welcome to check our paper and poster for more details.


